On Theorem Proving for Quantified Modal Logics:
With Applications in Modeling Ethical Principles

Naveen Sundar Govindarajulu and Selmer Bringsjord

Rensselaer AI & Reasoning Laboratory,
naveensundarg@gmail.com • selmer.bringsjord@gmail.com

1 Introduction

We will present and discuss two different algorithms, and lessons in their implementations, for carrying out reasoning efficiently and accurately in the $\mathcal{DCEC}$ (deontic cognitive event calculus) family of quantified modal logics. These logics have been used to model ethical theories and principles [8, 6, 9, 7, 4]. Both algorithms proceed by building reductions to first-order logic.[1] We will discuss advantages and disadvantages of our system with other systems for theorem proving in quantified modal logics. During the workshop, we will compare and contrast with existing systems by Benzmüller and colleagues [1, 2] and present initial results on benchmarking the above two algorithms and their implementations.

2 Two High-level Algorithms

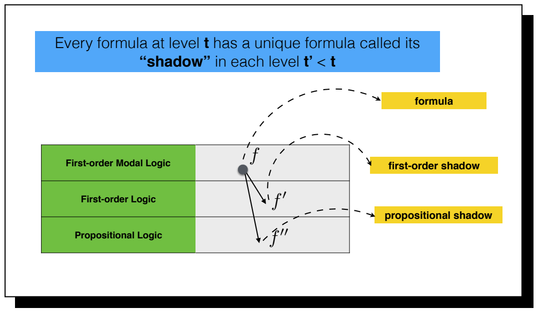
Algorithm I The first algorithm uses a technique called **shadowing** to achieve speed without sacrificing consistency in the system. Extant first-order modal logic theorem provers that can work with arbitrary inference schemata are built upon first-order theorem provers. They achieve the reduction to first-order logic via two modes. In the first mode, modal operators are simply represented by first-order predicates. This approach is the fastest but can quickly lead to well-known inconsistencies as demonstrated in [3]. In the second mode, the entire proof theory is implemented intricately in first-order logic, and the reasoning is carried out within first-order logic. Here, the first-order theorem prover simply functions as a declarative programming system. This approach, while accurate, can be excruciatingly slow. We use a different approach, in which we alternate between calling a first-order theorem prover and applying modal inference schemata. When we call the first-order prover, all modal atoms are converted into propositional atoms (i.e., shadowing), to prevent substitution into modal contexts. This approach achieves speed without sacrificing consistency. The prover also lets us add arbitrary inference schemata to the calculus by using a special-purpose language. The algorithm is briefly described below.

First we define the syntactic operation of **atomizing** a formula denoted by A. Given any arbitrary formula ϕ, $\mathsf{A}_{[\phi]}$ is a unique atomic (propositional) symbol. Next, we define the **level** of a formula: $\mathsf{level} : \mathsf{Boolean} \rightarrow \mathbb{N}$.

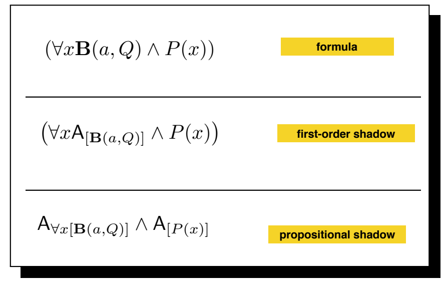
$$\mathsf{level}(\phi) = \begin{cases} 0; \phi \text{ is purely propositional formulae; e.g. } \textit{Rainy} \\ 1; \phi \text{ has first-order predicates or quantifiers e.g. } \textit{Sleepy}(\textit{jack}) \\ 2; \phi \text{ has modal formulae e.g. } \mathbf{K}(a, t, \textit{Sleepy}(\textit{jack})) \end{cases}$$

Given the above definition, we can define the operation of **shadowing** a formula to a level. See Figures 1a and 1b.

[1] Implementations are available at: https://github.com/naveensundarg/prover.



(a) Shadowing a formulae to different levels (Overview)



(b) Shadowing a formulae to different levels (Examples)

Shadowing

To shadow a formula χ to a level l, replace all sub-formulae χ' in χ such that level$(\chi') > l$ with $\mathsf{A}_{[\chi']}$ simultaneously. We denote this by $\mathsf{S}[\phi, l]$.
For a set Γ, the operation of shadowing all members in the set is simply denoted by $\mathsf{S}[\Gamma, l]$.

Assume we have access to a first-order prover $\mathbf{P}_F$. For a set of pure first-order formulae Γ and a first-order ϕ, $\mathbf{P}_F(\Gamma, \phi)$ gives us a proof of $\Gamma \vdash \phi$ if such a first-order proof exists, otherwise returns **fail**. See the algorithm sketch given below for Algorithm I:

Input: Input Formulae Γ, Goal Formula ϕ
Output: A proof of $\Gamma \vdash \phi$ if such a proof exists, otherwise **fail**
initialization;
while *goal not reached* **do**
  $answer = \mathbf{P}_F(\mathsf{S}[\Gamma, 1], \mathsf{S}[\phi, 1])$;
  if $answer \neq$ **fail** **then**
    return *answer* ;
  else
    $\Gamma' \longleftarrow$ **expand** Γ by using any applicable modal rules;
    if $\Gamma' = \Gamma$ **then**
      `/* The input cannot be expanded further */`
      return **fail**
    else
      set $\Gamma \longleftarrow \Gamma'$
    end
  end
end

Algorithm II The proof calculus for $\mathcal{DCEC}$ can be considered to be extension of standard first-order proof calculus under different *modal contexts*. For example, if a believes that b believes in a set of propositions Γ and $\Gamma \vdash_{FOL} Q$, then a believes that b believes Q. We convert $\mathbf{B}(a, t_a, \mathbf{B}(b, t_b, Q))$ into the pure first-order formula $Q(context(a, t_a, b, t_b))$ and use a first-order prover. The conversion process is a bit more nuanced as we have to convert compound formulae within iterated beliefs into proper conjunctive normal form clauses.

3 User Interface

The theorem prover is available as an open source Java library [10]. The prover is also integrated within the HyperSlate proof assistant, which is a modern extension of the Slate proof assistant [5]. See Figure 2

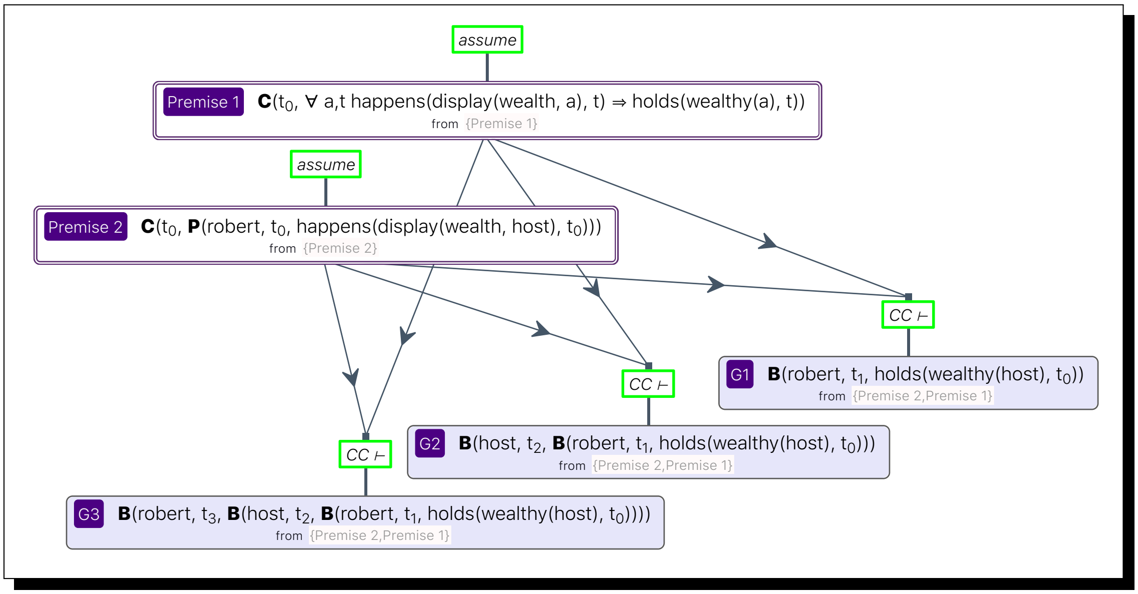


Figure 2: Use of the $\mathcal{DCEC}$ theorem prover within the HyperSlate workspace. Example from [**?**]

for an example.

References

[1] Christoph Benzmüller and Bruno Woltzenlogel Paleo. Automating Gödel's Ontological Proof of God's Existence with Higher-order Automated Theorem Provers. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the European Conference on Artificial Intelligence 2014 (ECAI 2014)*, pages 93–98, Amsterdam, The Netherlands, 2014. IOS Press.

[2] Christoph Benzmuüller and Bruno Woltzenlogel Paleo. The Inconsistency in Goödel's Ontological Argument: A Success Story for AI in Metaphysics. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pages 936–942. AAAI Press, 2016.

[3] S. Bringsjord and N. S. Govindarajulu. Given the Web, What is Intelligence, Really? *Metaphilosophy*, 43(4):361–532, 2012. This URL is to a preprint of the paper.

[4] Selmer Bringsjord, Naveen Sundar Govindarajulu, Daniel Thero, and Mei Si. Akratic Robots and the Computational Logic Thereof. In *Proceedings of ETHICS • 2014 (2014 IEEE Symposium on Ethics in Engineering, Science, and Technology)*, pages 22–29, Chicago, IL, 2014. IEEE Catalog Number: CFP14ETI-POD.

[5] Selmer Bringsjord, Joshua Taylor, Andrew Shilliday, Micah Clark, and Konstantine Arkoudas. Slate: An Argument-Centered Intelligent Assistant to Human Reasoners. In Floriana Grasso, Nancy Green, Rodger Kibble, and Chris Reed, editors, *Proceedings of the 8th International Workshop on Computational Models of Natural Argument (CMNA 8)*, pages 1–10, Patras, Greece, July 21 2008. University of Patras.

[6] Naveen Sundar Govindarajulu, Selmer Bringsjord, and Rikhiya Ghosh. One Formalization of Virtue Ethics via Learning. *CoRR*, abs/1805.07797, 2018. To be Presented at the 2018 International Association for Computing and Philosophy (IACAP) - Annual Meeting, June 21-23, 2018, Warsaw, Poland.

[7] Naveen Sundar Govindarajulu, Selmer Bringsjord, Rikhiya Ghosh, and Matthew Peveler. Beyond the doctrine of double effect: A formal model of true self-sacrifice. Proceedings of the 2017 International Conference on Robot Ethics and Safety Standards, forthcoming.

[8] Naveen Sundar Govindarajulu, Selmer Bringsjord, Rikhiya Ghosh, and Vasanth Sarathy. Towards the engineering of virtuous machines. In *Proceedings of the 2nd AAAI/ACM Workshop on Artificial Intelligence, Ethics, and Society (AIES-19)*, 2019.

[9] N.S. Govindarajulu and S. Bringsjord. On Automating the Doctrine of Double Effect. In C. Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 4722–4730. International Joint Conferences on Artificial Intelligence, 2017.

[10] Govindarajulu, Naveen Sundar. ShadowProver, 2016.